

# Team AWFY: Analyzing Massive Social Graphs *in a Blink of an Eye*

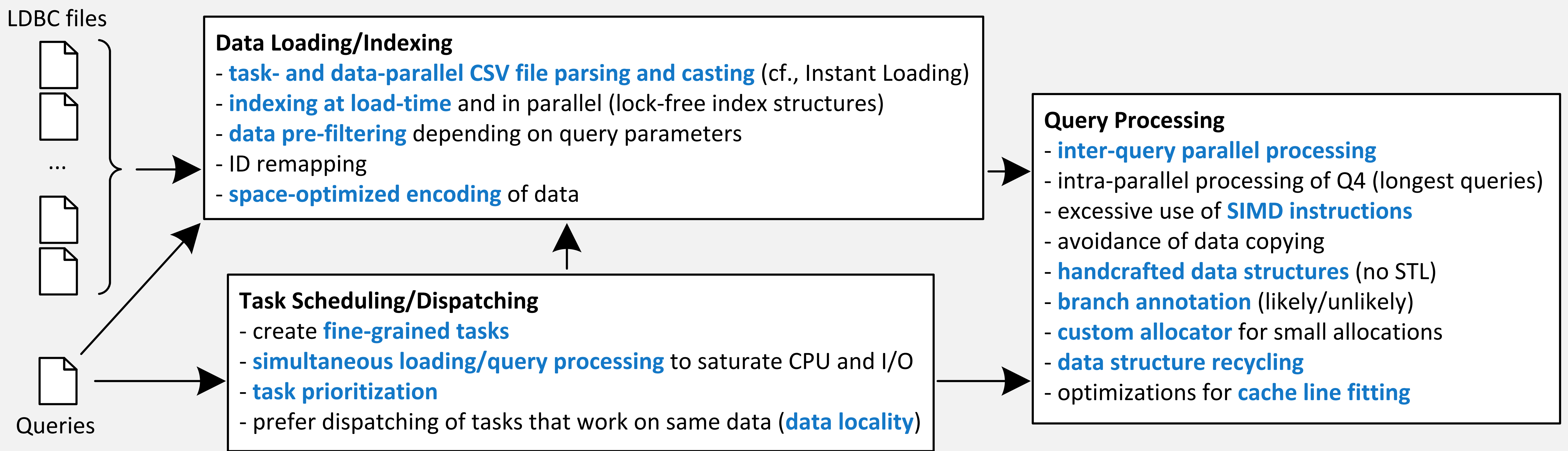
Moritz Kaufmann, Manuel Then, Tobias Mühlbauer, Andrey Gubichev  
{kaufmanm,then,muehlbau,gubichev}@in.tum.de

## SIGMOD 2014 Contest Challenge

- Task:** Implementation of a social network analysis system
- Input:** 31 CSV files generated by the LDBC (Linked Data Benchmark Council) social network benchmark dataset generator
- Query Types:**
- Q1) Shortest Distance Over Frequent Communication Paths
  - Q2) Interests with Large Communities
  - Q3) Socialization Suggestion
  - Q4) Most Central People

Rank	Team	Small (sec)	Medium (sec)	Large (sec)
1	 AWFY	0.126	0.237	2.849
2	unknown	0.124	0.331	3.674
3	 VIDA	0.207	0.469	4.571
4	 H_minor_free (The University of Tokyo)	0.127	0.366	6.046
5	 VSB_TUO (VSB)	0.185	0.699	6.824
6	 Bolzano_Nguyen	0.127	0.497	9.833
7	bxlrsm	0.126	0.555	9.852
8	Cracker	0.193	0.765	12.276
9	 UCY_YouSeeWhy (DMSL_University of Cyprus)	0.228	1.193	14.255
10	 GGWP (University of Cyprus)	0.471	2.372	28.689

## Overview of our Approach



### Q1

- **pre-calculate** person-replied-person relationship
- **bi-directional BFS** on index
- **batch Q1** for high throughput

### Q2

- **pre-calculate** youngest person and total number of persons per interest
- **prune by interest/by age**
- compute connected components using BFS

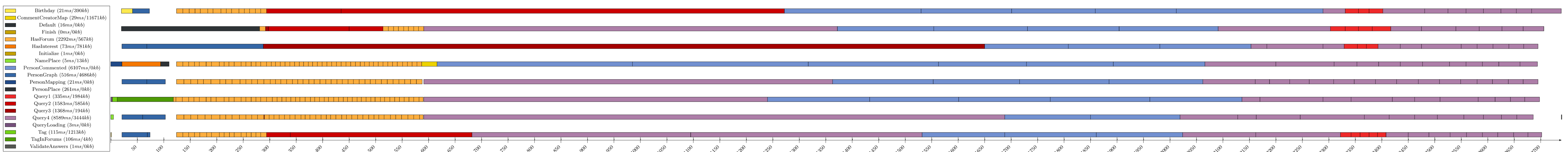
### Q3

- **location lookups using nested interval encoding**
- highly tuned **set intersection using SIMD operations**
- optimized BFS

### Q4

- **intelligent pruning strategy**
- **selective forum member loading**
- highly tuned BFS
- **intra-query parallelization**
- **advanced estimates**

## Execution Graph: LDBC 100k



## Code Sample: Count Lines

```
do {
    uint16_t a;
    __m128i data1=__mm_loadu_si128(reinterpret_cast<const __m128i*>(it));
    __m128i data2=__mm_loadu_si128(reinterpret_cast<const __m128i*>(it+16));
    a.i16[0]=__mm_movemask_epi8(__mm_cmpeq_epi8(data1,separator));
    a.i16[1]=__mm_movemask_epi8(__mm_cmpeq_epi8(data2,separator));
    lines+=__builtin_popcount(a.i32);
    it+=32;
} while (it<limit);
```

## Code Statistics

- **Files:** 313
- **Languages:** C++/make/shell
- **Lines of code:** 35 075
- **Lines of comments:** 5 426
- **git commits:** 742
- **average commits per day:** 6.8
- **Lines of text in repository:** 50 817 (90 304 added, 39 487 removed)